

WHAT IS CLAIMED IS:

1 1. A method comprising:

2 processing a definition of a function associated with a first language to create description
3 information about the function, the description information being sufficient to enable translation
4 of a call to the function into a call to a corresponding function in a second language without
5 requiring processing of the definition of the function.

6
1 2. The method of claim 1, further comprising:

2 storing the description information in a file of description items.

3
1 3. The method of claim 1, wherein processing the definition of the function

2
2.1 comprises:

3
3.1 examining the definition of the function associated with the first language;

4 deriving information about the function; and

5 using the derived information to translate the call to the function into a call to a
corresponding function in the second language.

6
1 4. The method of claim 3, further comprising:

2 using the derived information to create the description information.

3
1 5. The method of claim 3, further comprising:

2 storing the translated function in the second language in a library of entries.

3
1 6. The method of claim 1, in which processing the definition of the function

2 comprises:

3 deriving a number of declared formal inputs to the function.

4
1 7. The method of claim 1, in which processing the definition of the function

2 comprises:

3 deriving a number of declared formal outputs to the function.

1 8. The method of claim 1, in which processing the definition of the function
2 comprises:

3 deriving a scope of the function.

1 9. The method of claim 1, in which processing the definition of the function
2 comprises:

3 determining whether the function accepts a variable number of arguments.

1 10. The method of claim 1, in which processing the definition of the function
2 comprises:

3 determining whether the function returns a variable number of results.

4 11. A method comprising:

5 providing a file of description items, each item including description information about a
6 function associated with a first language, the description information being sufficient to enable
7 translation of a call to the function into a call to a corresponding function in a second language
8 without requiring processing of the definition of the function; and

9 using the file of description items to translate a first program file into a second program
10 file.

11 12. The method of claim 11, wherein the description information about the function
12 comprises:

13 a descriptor identifying a declared number of formal inputs to the function.

14 13. The method of claim 11, wherein the description information about the function
15 comprises:

16 a descriptor identifying a declared number of formal outputs to the function.

17 14. The method of claim 11, wherein the description information about the function
18 comprises:

19 a descriptor identifying a scope of the function.

1 15. The method of claim 11, wherein the description information about the function
2 comprises:

3 a descriptor identifying an acceptance of a variable input argument list into the function.

4
1 16. The method of claim 11, wherein the description information about the function
2 comprises:

3 a descriptor identifying a return of a variable output argument list from the function.

4
1 17. The method of claim 11, wherein using the file of description items comprises:
2 for each call to a function in the first program file, retrieving an item from the file of
3 description items;

4 using the description information in the item to translate the call to the function in the
5 first language into a call to a corresponding function in the second language; and
6 storing the translated function in the second program file.

7
1 18. The method of claim 11, wherein using the file of description items comprises:
2 generating a call through a function evaluation interface for the function if the description
3 information includes a descriptor identifying an acceptance of a variable input argument list into
4 the function.

5
1 19. The method of claim 11, wherein using the file of description items comprises:
2 generating a call through a function evaluation interface for the function if the description
3 information includes a descriptor identifying a return of a variable output argument list from the
4 function.

6
1 20. The method of claim 11, wherein using the file of description items comprises:
2 generating a call through a normal interface for the function if the description information
3 includes a descriptor identifying a known number of input and output arguments to the function.

7
1 21. A method comprising:

2 providing a library file including functions defined by a first language;

3 processing the library file to create a function library and a description file, the function
4 library including one or more functions defined by a second language, each function in the
5 function library being a translated version of a function in the library file, and the description file
6 including description information about each function in the library file, the description
7 information being sufficient to enable translation of a call to the function into a call to a
8 corresponding function in the second language without requiring processing of the definition of
9 the function; and

10 using the description file to translate a program file from the first language into the
11 second language, wherein each call in the program file to a function in the library file is
12 translated into a call to a corresponding function in the second language.

13

1 22. The method of claim 21, wherein processing the library file comprises:
2 translating the call to each function in the library file into a call to a corresponding
3 function in the second language; and
4 creating a function library including the translated version of each function in the library
5 file.

6

1 23. The method of claim 22, further comprising:
2 examining the definition of each function in the library file;
3 deriving information about each function; and
4 using the derived information to translate the call to each function into a call to a
5 corresponding function in the second language.

6

1 24. The method of claim 23, further comprising:
2 using the derived information about each function to create the description information;
3 and
4 creating a description file including description information about each function in the
5 library file.

6

7

8

1 25. The method of claim 21, wherein using the description file comprises:
2 for each call in the program file to a function in the library file, retrieving the description
3 information about the function from the description file; and
4 using the description information to translate the call to the function in the first language
5 into a call to a corresponding function in the second language.

6
1 26. The method of claim 21, wherein using the description file comprises:
2 generating a call through a function evaluation interface for the function if the description
3 information includes a descriptor identifying an acceptance of a variable input argument list into
4 the function.

5
1 27. The method of claim 21, wherein using the description file comprises:
2 generating a call through a function evaluation interface for the function if the description
3 information includes a descriptor identifying a return of a variable output argument list from the
4 function.

5
1 28. The method of claim 21, wherein using the description file comprises:
2 generating a call through a normal interface for the function if the description information
3 includes a descriptor identifying a known number of input and output arguments to the function.

5
1 29. A computer program product, tangibly stored on a computer-readable medium,
2 for creating a data file, the product comprising instructions operable to cause a programmable
3 processor to:

4 process a definition of a function associated with a first language to create description
5 information about the function, the description information being sufficient to enable translation
6 of a call to the function into a call to a corresponding function in a second language without
7 requiring processing of the definition of the function.

8
1 30. The product of claim 29, further comprising instructions operable to cause a
2 programmable processor to:
3 store the description information in a file of description items.

1 31. The product of claim 29, wherein processing the definition of the function
2 comprises:

3 examining the definition of the function associated with the first language;
4 deriving information about the function; and
5 using the derived information to translate the call to the function into a call to a
6 corresponding function in the second language.

7
1 32. The product of claim 31, further comprising instructions operable to cause a
2 programmable processor to:

3 use the derived information to create the description information.

4
1 33. The product of claim 31, further comprising instructions operable to cause a
2 programmable processor to:

3 store the translated function in the second language in a library of entries.

4
1 34. The product of claim 29, in which processing the definition of the function
2 comprises:

3 deriving a number of declared formal inputs to the function.

4
1 35. The product of claim 29, in which processing the definition of the function
2 comprises:

3 deriving a number of declared formal outputs to the function.

4
1 36. The product of claim 29, in which processing the definition of the function
2 comprises:

3 deriving a scope of the function.

4
1 37. The product of claim 29, in which processing the definition of the function
2 comprises:

3 determining whether the function accepts a variable number of arguments.

1 38. The product of claim 29, in which processing the definition of the function
2 comprises:

3 determining whether the function returns a variable number of results.

4
1 39. A product, stored on a machine-readable medium, for translating a program file,
2 the product comprising instructions operable to cause a processor to:

3 provide a file of description items, each item including description information about a
4 function associated with a first language, the description information being sufficient to enable
5 translation of a call to the function into a call to a corresponding function in a second language
6 without requiring processing of the definition of the function; and

7 use the file of description items to translate a first program file into a second program
8 file.

9
1 40. The product of claim 39, wherein the description information about the function
2 comprises:

3 a descriptor identifying a declared number of formal inputs to the function.

4
1 41. The product of claim 39, wherein the description information about the function
2 comprises:

3 a descriptor identifying a declared number of formal outputs to the function.

4
1 42. The product of claim 39, wherein the description information about the function
2 comprises:

3 a descriptor identifying a scope of the function.

4
1 43. The product of claim 39, wherein the description information about the function
2 comprises:

3 a descriptor identifying an acceptance of a variable input argument list into the function.

1 44. The product of claim 39, wherein the description information about the function
2 comprises:

3 a descriptor identifying a return of a variable output argument list from the function.

4
1 45. The product of claim 39, wherein using the file of description items comprises:
2 for each call to a function in the first program file, retrieving an item from the file of
3 description items;

4 using the description information in the item to translate the call to the function in the
5 first language into a call to a corresponding function in the second language; and
6 storing the translated function in the second program file.

7
1 46. The product of claim 39, wherein using the file of description items comprises:
2 generating a call through a function evaluation interface for the function if the description
3 information includes a descriptor identifying an acceptance of a variable input argument list into
4 the function.

5
1 47. The product of claim 39, wherein using the file of description items comprises:
2 generating a call through a function evaluation interface for the function if the description
3 information includes a descriptor identifying a return of a variable output argument list from the
4 function.

6
1 48. The product of claim 39, wherein using the file of description items comprises:
2 generating a call through a normal interface for the function if the description information
3 includes a descriptor identifying a known number of input and output arguments to the function.

4
1 49. A computer program product, tangibly stored on a computer-readable medium,
2 for translating function calls, the product comprising instructions operable to cause a
3 programmable processor to:
4 provide a library file including functions defined by a first language;
5 process the library file to create a function library and a description file, the function
6 library including one or more functions defined by a second language, each function in the

7 function library being a translated version of a function in the library file, and the description file
8 including description information about each function in the library file, the description
9 information being sufficient to enable translation of a call to the function into a call to a
10 corresponding function in the second language without requiring processing of the definition of
11 the function; and

12 use the description file to translate a program file from the first language into the second
13 language, wherein each call in the program file to a function in the library file is translated into a
14 call to a corresponding function in the second language.

15

1 50. The product of claim 49, wherein processing the library file comprises:

2 translating the call to each function in the library file into a call to a corresponding
3 function in the second language; and

4 creating a function library including the translated version of each function in the library
5 file.

6 51. The product of claim 49, further comprising:

7 examining the definition of each function in the library file;

8 deriving information about each function; and

9 using the derived information to translate the call to each function into a call to a
10 corresponding function in the second language.

11 52. The product of claim 51, further comprising:

12 using the derived information about each function to create the description information;
13 and

14 creating a description file including description information about each function in the
15 library file.

16

17

18 53. The product of claim 49, wherein using the description file comprises:

19 for each call in the program file to a function in the library file, retrieving the description
20 information about the function from the description file; and

4 using the description information to translate the call to the function in the first language
5 into a call to a corresponding function in the second language.

6
1 54. The product of claim 49, wherein using the description file comprises:
2 generating a call through a function evaluation interface for the function if the description
3 information includes a descriptor identifying an acceptance of a variable input argument list into
4 the function.

5
1 55. The product of claim 49, wherein using the description file comprises:
2 generating a call through a function evaluation interface for the function if the description
3 information includes a descriptor identifying a return of a variable output argument list from the
4 function.

56. The product of claim 49, wherein using the description file comprises:
 generating a call through a normal interface for the function if the description information
includes a descriptor identifying a known number of input and output arguments to the function.